Solving Elliptic PDEs on Overlapping Grids with Multigrid

Bill Henshaw


Centre for Applied Scientific Computing,
Lawrence Livermore National Laboratory,
Livermore, CA, 94551,
henshaw@llnl.gov

Summary: Black-Box Multigrid for Overlapping Grids

1. Automatic generation of coarse grid levels and coarse grid interpolation points.

2. Generation of *any* number of multigrid levels (as allowed by the number of grid points).

3. Automatic generation of coarse grid equations through Galerkin averaging of fine grid equations.

4. Optimised for pre-defined equations (such as the Laplace and Heat operator) and optimised for cartesian grids (speed and memory usage).

Problem Specification

Suppose that we wish to solve an elliptic boundary value problem on a domain $\Omega \subset R^d$ given by

$$
\begin{aligned}
Lu &= f \qquad \mathbf{x} \in \Omega, \\
Bu &= g \qquad \mathbf{x} \in \Gamma = \partial\Omega.
\end{aligned}
$$

We can discretize these equations on an overlapping grid resulting in a set of discrete equations,

$$
\begin{aligned}
L_h u_h &= f_h \qquad \mathbf{x}_\mathbf{i} \in \Omega_h \\
B_h u_h &= g_h \qquad \mathbf{x}_\mathbf{i} \in \Gamma_h \\
I_h u_h &= 0 \qquad \mathbf{x}_\mathbf{i} \in \Gamma_h^I \text{ (interpolation)}
\end{aligned}
$$

It is easy to coarsen each component grid of an overlapping grid. It is more difficult to determine interpolation points on the coarse grids.

Background

The first overlapping grid computations were apparently performed by G. Starius, a student of Heinz-Otto Kreiss, who solved elliptic and hyperbolic problems [6, 7].

The first MG solver for overlapping grids seems to be the work of J. Linden reported in Stüben and Trottenberg [8] who showed results for a model problem.

Chesshire and Henshaw [3] extended the CMPGRD overlapping grid generator [1] to generate multigrid levels for general two-dimensional domains. These grids were used to solve elliptic problems in two dimensions for general domains and showed good multigrid convergence rates.

Due to the difficulty in generating the interpolation equations to couple the equations on the coarse grids, most if not all other researchers have left the coarse grids uncoupled, applying a zero dirichlet or neumann type boundary condition at interpolation points, Tu & Fuchs [9, 10], Hinatsu & Ferziger [4], Zang & Street [11]. This approach has been called **incomplete multigrid** (ICMG) by Hinatsu & Ferziger. In general it would seem that ICMG can converge no better than an overlapping Schwartz iteration with a convergence rate $1 - O(\delta)$ where δ is the relative width of the overlap.

Multigrid Operators

The fundamental structure of the multigrid algorithm for overlapping grids remains the same as for a single grid. Introduce the following operators

$\mathbf{S}$: the composite smoothing operator, an iteration that approximately solves the equation and is effective at reducing the high frequency components of the error.

$\mathbf{R}_k^{k-1}$: restriction operator, the operator that transfers a grid function from the fine grid to the coarse grid.

$\mathbf{P}_k^{k+1}$: prolongation operator, the operator that transfers a grid function from the coarse grid to the fine grid.

Multigrid Algorithm

while *not converged* **do**

smooth ν_1 times

$v_1 \leftarrow \mathbf{S}^{\nu_1} v_1$

form the defect and transfer to the coarser grid

$f_2 \leftarrow \mathbf{R}_1^2(f_1 - A_1 v_1)$

"solve" the defect equation

$A_2 v_2 \approx f_2$

correct the fine grid solution from the coarse grid solution

$v_1 \leftarrow v_1 + \mathbf{P}_2^1 v_2$

smooth ν_2 times

$v_1 \leftarrow \mathbf{S}^{\nu_2} v_1$

end while

The coarse grid equations can be approximately solved in a recursive many by using an even coarser grid. On the very coarsest grid the equations are solved with a sparse matrix solver using either an iterative or direct method.

Composite smoothing operator

for $g=1,\ldots,n_g$ **do** (loop over component grids)

  if $g>1$

    interpolate points on grid G_g

  end if

  for $m=1,\ldots,\nu_g$ (multiple sub-smooths)

    if $m<\nu_g$

      where mask>0

        $u \leftarrow u+\omega(Lu-f)$ (do not change interpolation pts)

      end where

    else

      $u \leftarrow u+\omega(Lu-f)$

    end if

    apply boundary conditions to grid G_g

  end for

end for

interpolate all points on the overlapping grid $\mathcal{G}$.

Composite residual operator

A valid defect for the interior PDE can be computed at all discretization points but not at interpolation points. In some cases a discretization point on the coarse grid point will, using the full-weighting operator, require a defect to be defined at the interpolation point. We therefore provide a value of the defect at interpolation points that approximates the defect at nearby interpolation points. We have evaluated two different ways of treating this case:

1. interpolate the fine grid defect from other grids to determine the defect values at the interpolation points.
2. obtain values at the interpolation points by extrapolating the values from nearby discretization points.

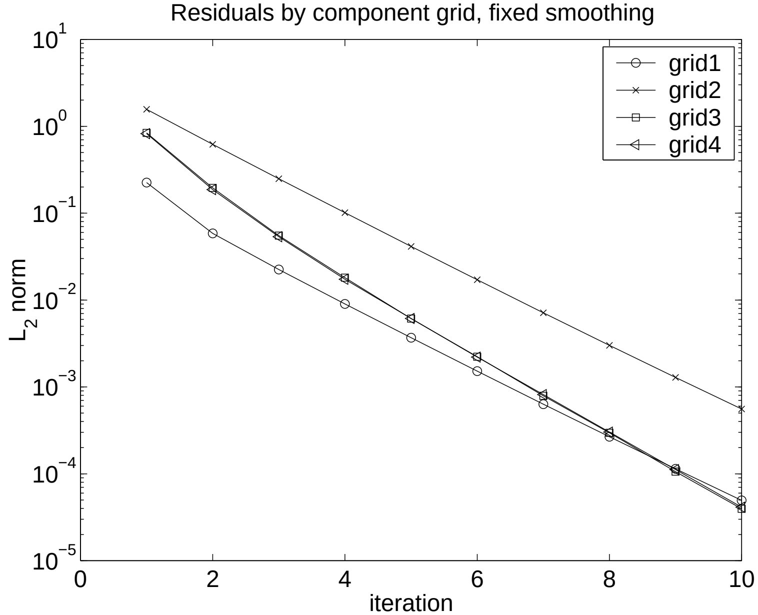
We have found that interpolating the defect seems to produce the best results.

In order to measure the convergence rate we introduce the following notation:

$$
\begin{aligned}
\text{WU(i)} &= \text{number of work units for iteration i} \ , \\
\text{res(i)} &= \text{residual for iteration i (maximum norm)} \ , \\
\text{rate(i)} &= \text{convergence rate (CR), res(i)/res(i-1)} \ , \\
\text{ECR(i)} &= \text{effective convergence rate} \ , \\
&= (\text{rate}(i))^{1/WU(i)} \ , \\
\text{V(m,n)} &= \text{denotes a V cycle with m pre-smooths and n post-smooths} \ .
\end{aligned}
$$

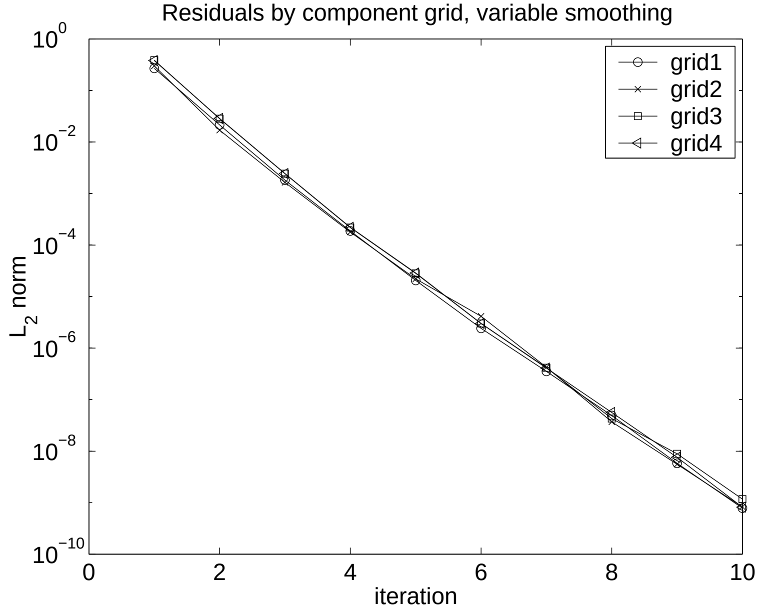
A work unit is defined to be the amount of work (number of multiplications) required for a single Jacobi iteration. The work units reported here are only reasonable approximations.

i	res	rate	WU	ECR
1	$6.5e+00$	0.161	4.7	0.68
2	$2.6e+00$	0.402	4.7	0.82
3	$1.1e+00$	0.431	4.7	0.83
4	$4.8e-01$	0.430	4.7	0.83
5	$2.1e-01$	0.432	4.7	0.84
6	$9.0e-02$	0.433	4.7	0.84
7	$3.9e-02$	0.435	4.7	0.84
8	$1.7e-02$	0.437	4.7	0.84
9	$7.5e-03$	0.439	4.7	0.84
10	$3.3e-03$	0.441	4.7	0.84



Fixed number of sub-smooths per grid

i	res	rate	WU	ECR
1	$3.3e+00$	0.081	6.1	0.66
2	$3.7e-01$	0.113	7.0	0.73
3	$3.2e-02$	0.087	6.5	0.69
4	$3.2e-03$	0.101	6.1	0.69
5	$3.7e-04$	0.113	5.9	0.69
6	$4.0e-05$	0.110	5.6	0.67
7	$6.6e-06$	0.165	6.1	0.74
8	$9.6e-07$	0.144	6.6	0.75
9	$1.0e-07$	0.109	6.1	0.70
10	$1.3e-08$	0.122	6.1	0.71



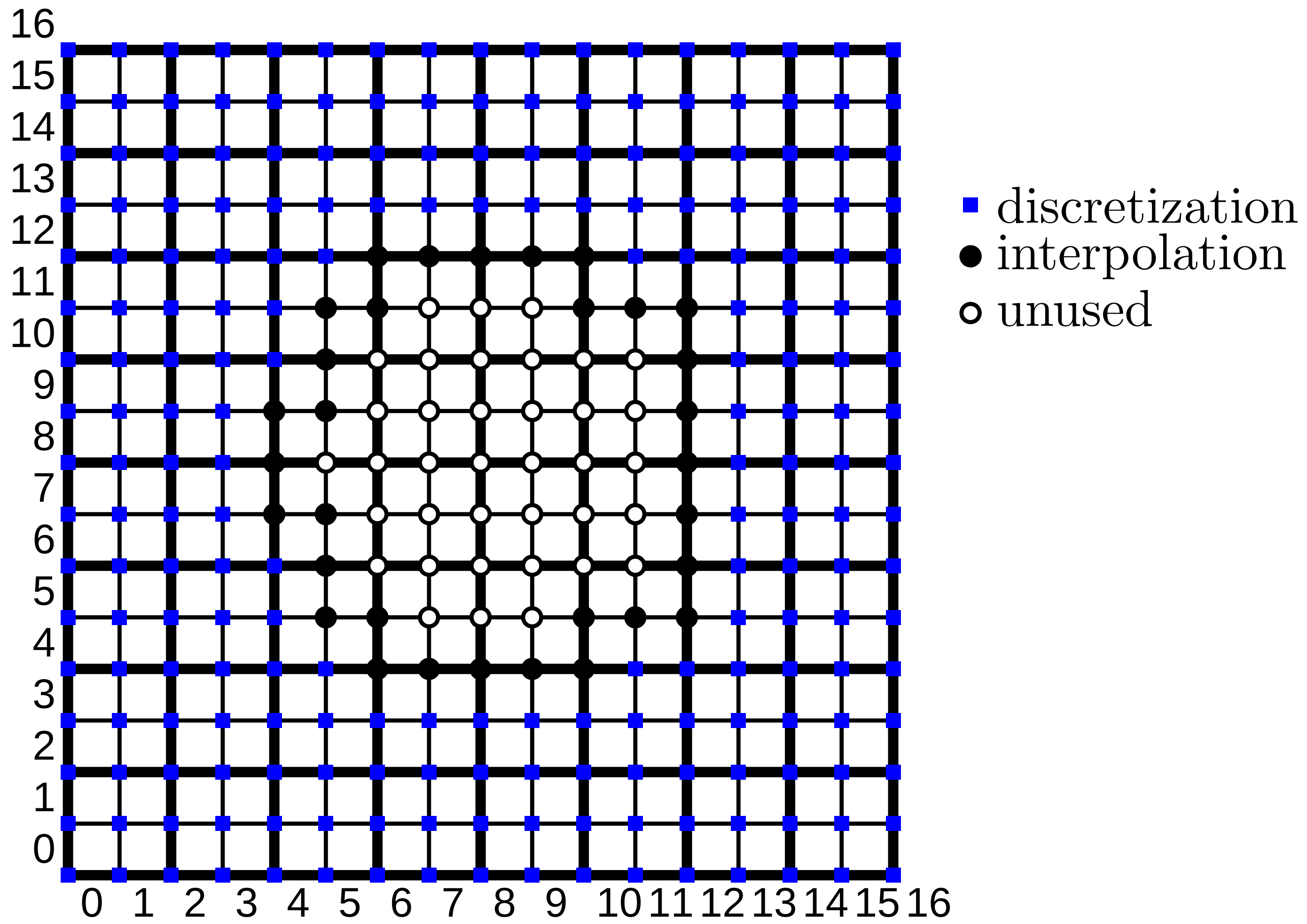
Variable number of sub-smooths per grid

Automatic Generation of Coarse Grid Levels

The key ingredients to the coarsening algorithm are as follows:

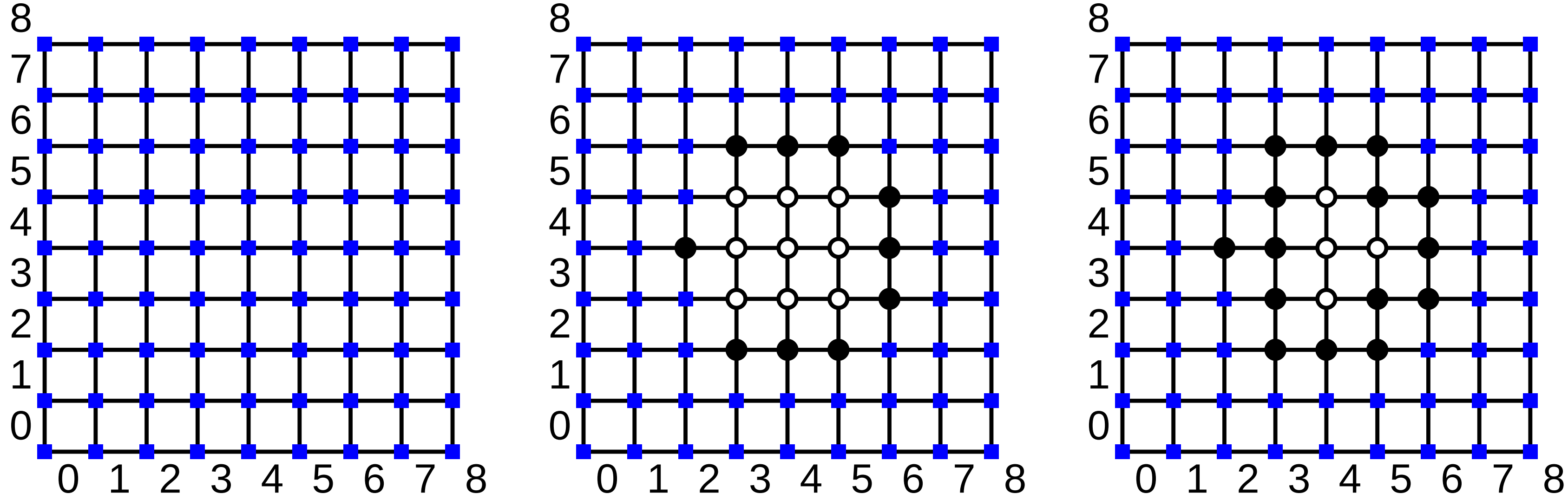
1. interpolate ghost points on interpolation boundaries. When a component grid is coarsened the new ghost points will be further from the boundary of the grid. As a result the effective overlap between neighbouring component grids will increase.
2. relax the accuracy and explicitness of the interpolation on coarse grids. As the grids are coarsened we
   (a) allow explicit interpolation to become implicit; implicit interpolation requires less overlap.
   (b) allow the width of the interpolation stencil to decrease; we thus allow each point to have a possibly different interpolation width.
   (c) allow a coarse grid interpolation point that has extended outside the domain to be set equal to the closes point on the boundary

Automatic Coarsening



Mask array for the fine grid.

Automatic Coarsening

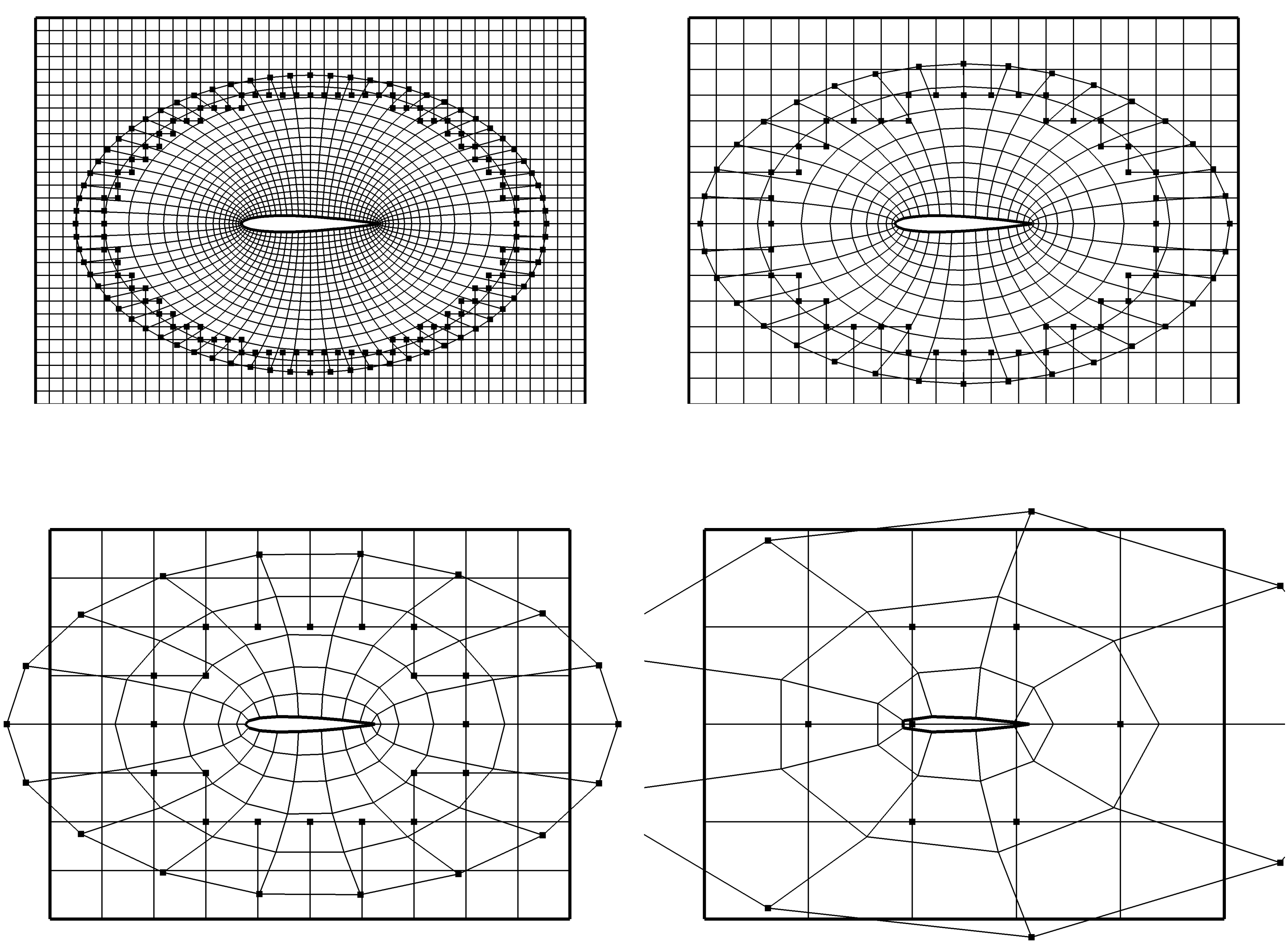


Mask array for the coarse grid.

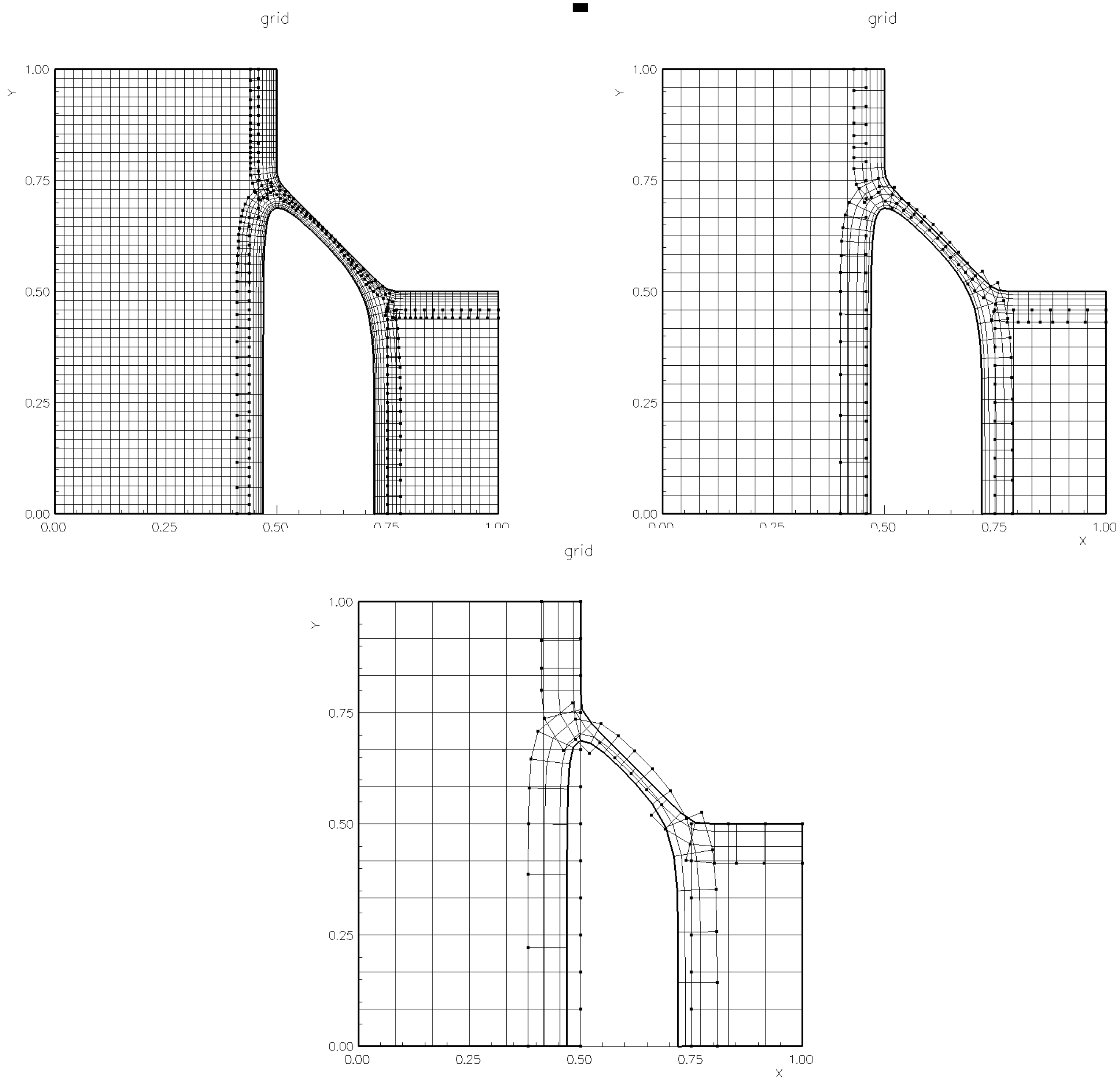
Left: initial state.

Middle: after assigning from fine grid mask.

Right: after filling in extra interpolation points.



An overlapping grid for an airfoil, 4 multigrid levels.



An overlapping grid for a valve, 3 multigrid levels.

Coarse Grid Equations Through Averaging

To automatically generate a coarse grid operator from a fine grid operator we can average the operator on the fine grid and then restrict the result to the coarse grid. This approach is known as Galerkin averaging. Given the fine grid operator L_k the Galerkin coarse grid operators are defined as

$$L_{k+1} := \mathbf{R}_k^{k+1} L_k \mathbf{P}_{k+1}^k$$

where $\mathbf{R}_{k+1}^k$ and $\mathbf{P}_k^{k+1}$ are restriction and prologation operators. These operators are often taken to be the same as those used in the multigrid cycle.

To illustrate the approach, and to show how the Galerkin averaging might be implemented, consider a one-dimensional problem discretized with a three point stencil,

$$a_i u_{i-1} + b_i u_i + c_i u_{i+1} = f_i \qquad i = 1, 2, \ldots$$

If we look at the stencil for rows $i-1, i, i+1$ arranged in a matrix then we get

$$\begin{array}{ccccc} a_{i-1}u_{i-2} & b_{i-1}u_{i-1} & c_{i-1}u_i & 0 & 0 \\ 0 & a_i u_{i-1} & b_i u_i & c_i u_{i+1} & 0 \\ 0 & 0 & a_{i+1}u_i & b_{i+1}u_{i+1} & c_{i+1}u_{i+2} \end{array}$$

If we replace row i by the weighted average of rows $i-1, i, i+1$ with weights α, β, α then we get the wide stencil

$$\alpha a_{i-1} u_{i-2} \quad (\alpha b_{i-1} + \beta a_i) u_{i-1} \quad (\alpha(c_{i-1} + a_{i+1}) + \beta b_i) u_i \quad (\alpha b_{i+1} + \beta c_i) u_{i-1} \quad \alpha c_{i+1} u_{i+2}$$

Typically we take $\alpha = 1/4$, and $\beta = 1/2$.

If we distribute the values at point $i-1$ using $u_{i-1} = \frac{1}{2}(u_{i-2} + u_i)$ and at point $i+1$ using $u_{i+1} = \frac{1}{2}(u_{i+2} + u_i)$ then we have a wide stencil only defined at points $i-2, i, i+2$,

$$\hat{a}_i \ u_{i-2} \qquad 0 \cdot u_{i-1} \qquad \hat{b}_i \ u_i \qquad 0 \cdot u_{i+1} \qquad \hat{c}_{i+2} \ u_{i+2}$$

$$\hat{a}_i = \alpha(a_{i-1} + \frac{1}{2}b_{i-1}) + \frac{1}{2}\beta a_i$$

$$\hat{b}_i = \alpha(\frac{1}{2}b_{i-1} + \frac{1}{2}b_{i+1} + c_{i-1} + a_{i+1}) + \beta(b_i + \frac{1}{2}a_i + \frac{1}{2}c_i)$$

$$\hat{c}_i = \alpha(c_{i+1} + \frac{1}{2}b_{i+1}) + \frac{1}{2}\beta c_i$$

The coarse grid operator is then defined as

$$a_i^c u_{i-1}^c + b_i^c u_i^c + c_i^c u_{i+1}^c = f_i^c \qquad i = 1, 2, \ldots$$

$$a_i^c = \hat{a}_{2i}$$

$$b_i^c = \hat{b}_{2i}$$

$$c_i^c = \hat{c}_{2i}$$

As an example of the averaging process consider the 5-point discretization of the Laplacian on a rectangular grid

$$A_0 = \begin{bmatrix} 0 & \beta & 0 \\ \alpha & -2(\alpha+\beta) & \alpha \\ 0 & \beta & 0 \end{bmatrix} \text{ (initial 5-point stencil)}$$

where $\alpha = 1/h_x^2$ and $\beta = 1/h_y^2$. Let A_m denote the stencil after m steps of averaging scaled by the factor 4^m.

$$A_0 = \begin{bmatrix} 0 & \beta & 0 \\ \alpha & -2(\alpha+\beta) & \alpha \\ 0 & \beta & 0 \end{bmatrix} \quad \text{(initial 5-point stencil)}$$

$$A_1 = \begin{bmatrix} \frac{1}{8}(\alpha+\beta) & \frac{3}{4}\beta - \frac{1}{4}\alpha & \frac{1}{8}(\alpha+\beta) \\ \frac{3}{4}\alpha - \frac{1}{4}\beta & -\frac{3}{2}(\alpha+\beta) & \frac{3}{4}\alpha - \frac{1}{4}\beta \\ \frac{1}{8}(\alpha+\beta) & \frac{3}{4}\beta - \frac{1}{4}\alpha & \frac{1}{8}(\alpha+\beta) \end{bmatrix} \quad \text{(scaled stencil after 1 averaging)}$$

$$A_2 = \begin{bmatrix} \frac{5}{32}(\alpha+\beta) & \frac{11}{16}\beta - \frac{5}{16}\alpha & \frac{5}{32}(\alpha+\beta) \\ \frac{11}{16}\alpha - \frac{5}{16}\beta & -\frac{11}{8}(\alpha+\beta) & \frac{11}{16}\alpha - \frac{5}{16}\beta \\ \frac{5}{32}(\alpha+\beta) & \frac{11}{16}\beta - \frac{5}{16}\alpha & \frac{5}{32}(\alpha+\beta) \end{bmatrix} \quad \text{(scaled stencil after 2 averaging)}$$

$$A_\infty = \begin{bmatrix} \frac{1}{6}(\alpha+\beta) & \frac{2}{3}\beta - \frac{1}{3}\alpha & \frac{1}{6}(\alpha+\beta) \\ \frac{2}{3}\alpha - \frac{1}{3}\beta & -\frac{4}{3}(\alpha+\beta) & \frac{2}{3}\alpha - \frac{1}{3}\beta \\ \frac{1}{6}(\alpha+\beta) & \frac{2}{3}\beta - \frac{1}{3}\alpha & \frac{1}{6}(\alpha+\beta) \end{bmatrix} \quad \text{(limiting scaled stencil)}$$

Ogmg: Overture's Overlapping Grid Multigrid Solver

Ogmg can be called through the generic solver interface Oges. The multigrid levels and coarse grids operators are built automatically.

```
CompositeGrid cg(...); // Get a grid from somewhere
CompositeGridOperators cgop(cg); // Define operators
Oges solver; // Define a solver
OgesParameters solverParameters; // Parameters for solver
solverParameters.set(multigrid); // Choose multigrid
solver.setOgesParameters(solverParameters);
solver.setGrid( cg );
// Choose a predefined equation:
solver.setEquationAndBoundaryConditions(laplaceEquation,cgop,bc, bcData );
realCompositeGridFunction u,f; // grid functions for solution and rhs
...
solver.solve(u,f); // solve Δu = f
```

grid	d	grid pts	n_g	BC	n_l	cycle	CR	ECR
square	$2D$	128^2	1	D		V(2,1)	.027	0.49
square	$2D$	256^2	1	D		V(2,1)	.028	0.49
square	$2D$	128^2	1	N	4	V(2,1) RB	.044	0.54
square	$2D$	128^2	1	M	4	V(2,1) RB	.029	0.50
annulus	$2D$	$9,657$	1	D	4	V(2,1) RB	.048	0.55
circle in a channel	$2D$	$86,130$	2	D	4	V(2,1) RB	.099	0.71
circle in a channel	$2D$	$86,130$	2	D	4	V(1,1) RB	.141	0.68
circle in a channel	$2D$	$86,130$	2	N	4	V(2,1) RB	.131?	0.71
airfoil	$2D$	$11,378$	2	D	4	V(1,1)RB/Z	.070	0.62
shapes	$2D$	$6,456$	4	D	3	V(2,1)RB/Z	.142	0.76
box	$3D$	32^3	1	D	4	V(2,1)	.071	0.57
box	$3D$	64^3	1	D	4		.079	0.58
box	$3D$	32^3	1	M	4	V(2,1)RB	.059	0.54
sphere in a box	$3D$	$72,519$	3	D	3	V(2,1)RB	.062	0.69
sphere in a box	$3D$	$72,519$	3	MD	3	V(2,1)RB	.053	0.68
ellipsoid in a box	$3D$	$116,620$	4	D	3	V(2,1)RB	.112	0.70
ellipsoid in a box	$3D$	$116,620$	4	D	3	V(2,1)RB/Z	.082	0.71
ellipsoid in a box	$3D$	$737,700$	4	D	4	V(2,1)RB	.145	0.76
5 spheres in a box	$3D$	$437,839$	11	D	3	V(2,1)RB	.081	0.73

Multigrid convergence rates for various grids. The number of component grids is n_g and the number of multigrid levels is n_l. Boundary conditions, shown in the column labeled BC, are D for Dirichlet, N for Neumann, M for mixed and DN for a mix of Dirichlet and Neumann. CR is the average convergence rate per cycle, ECR is the average effective convergence rate per cycle.

Results from Ogmg

problem	grid points	CR	cpu/cycle	Memory	reals/pt
circ in sq (2D)	1.1 million	.056	.93	48 M	5.2
two circles (2D)	6.3 million	.043	4.7	287 M	5.7
ellipsoid (3D)	4.7 million	.12	8.5	442 M	11.7
two spheres (3D)	10.2 million	.085	29.2	1420 M	17.4

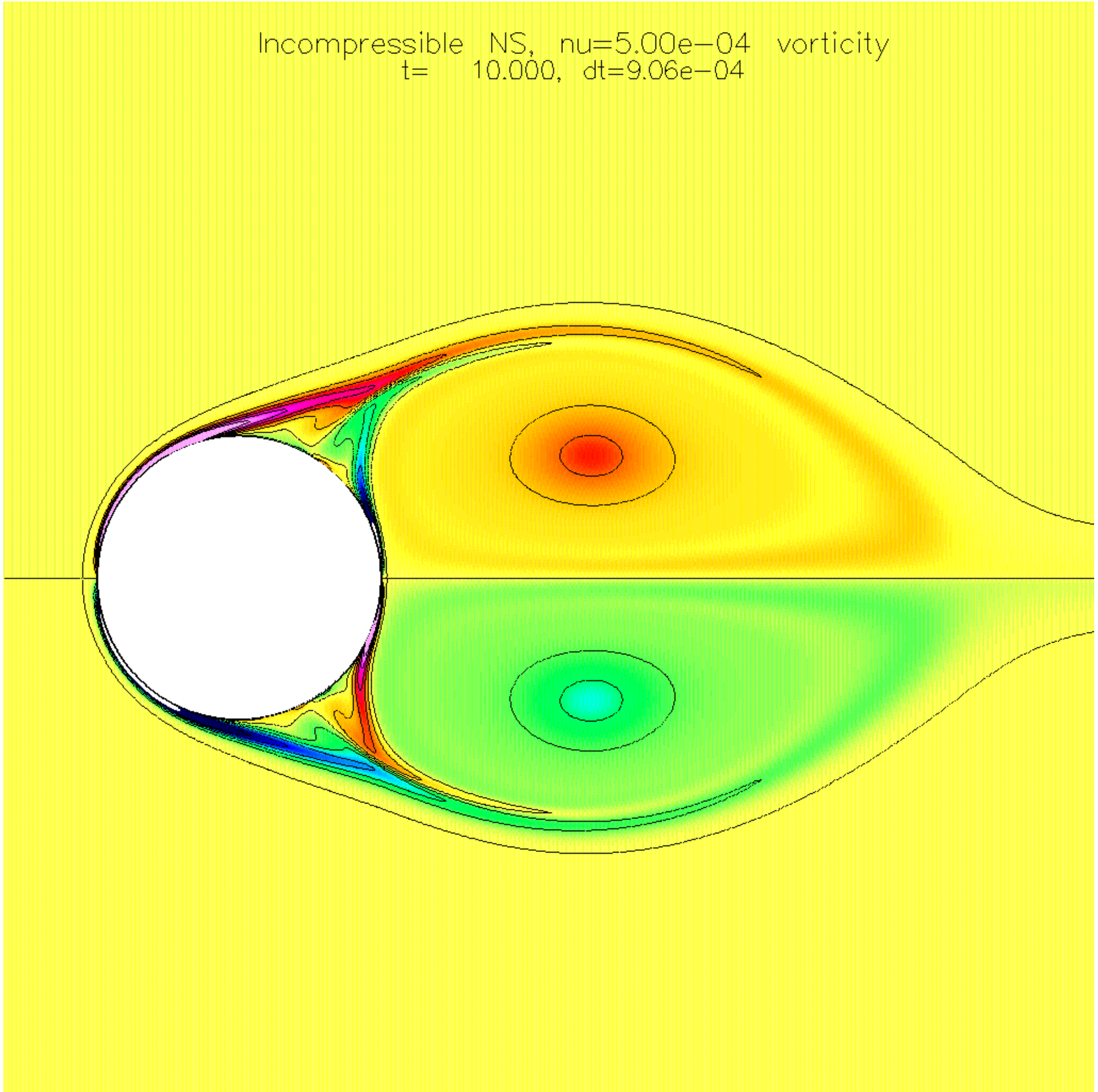
Performance and memory usage results for the predefined Laplace operator which has been optimised for cartesian grids; 2.2 GigaHertz Pentium workstation.

Results from Ogmg

Solver	grid	pts	its	res	CPU	setup	solve	reals/pt
Ogmg	cic	$1.1e6$	9	2.e-8	10.1	.92	9.2	5.2
PETSc	cic	$1.1e6$	1268	2.e-8	934.	43.	891.	26.5
Ogmg	ellipsoid	$7.4e5$	10	3.e-7	21.2	4.5	16.7	19.4
PETSc	ellipsoid	$7.4e5$	50	3.e-7	44.4	23.7	20.7	55.6

A comparison of the setup and solution times for multigrid and a Krylov space solver (bi-CG-stab) from PETSc

Incompressible Navier-Stokes Equations



Laminar flow past a cylinder. Multigrid is used for the pressure solve and implicit time-stepping. For 1.1 million grid points, OverBlown requires 320M, 7.6 s/step.

References

[1] G. Chesshire and W. Henshaw, *Composite overlapping meshes for the solution of partial differential equations*, J. Comp. Phys., 90 (1990), pp. 1–64.

[2] W. Henshaw, *Part II: Composite Overlapping Grid Techniques*, PhD thesis, Dept. of Applied Mathematics, California Institute of Technology, 1985.

[3] W. Henshaw and G. Chesshire, *Multigrid on composite meshes*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 914–923.

[4] M. Hinatsu and J. Ferziger, *Numerical computation of unsteady incompressible flow in complex geometry using a composite multigrid technique*, International Journal for Numerical Methods in Fluids, 13 (1991), pp. 971–997.

[5] R. A. Johnson and D. M. Belk, *Multigrid approach to overset grid communication*, AIAA J., 33 (1995), pp. 2305–2308.

[6] G. Starius, *Composite mesh difference methods for elliptic and boundary value problems*, Numer. Math., 28 (1977), pp. 243–258.

[7] ———, *On composite mesh difference methods for hyperbolic differential equations*, Numer. Math., 35 (1980), pp. 241–255.

[8] K. Stüben and U. Trottenberg, *Multigrid methods: Fundamental algorithms, model problem analysis and applications*, in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., Springer-Verlag, 1982, pp. 1–176.

[9] J. Y. Tu and L. Fuchs, *Overlapping grids and multigrid methods for the three-dimensional unsteady flow calculations in IC engines*, International Journal for Numerical Methods in Fluids, 15 (1992), pp. 693–714.

[10] ———, *Calculation of flows using three-dimensional overlapping grids and multigrid methods*, International Journal for Numerical Methods in Engineering, 38 (1995), pp. 259–282.

[11] Y. Zang and R. Street, *A composite multigrid method for calculating unsteady incompressible flows in geometrically complex domains*, International Journal for Numerical Methods in Fluids, 20 (1995), pp. 341–361.